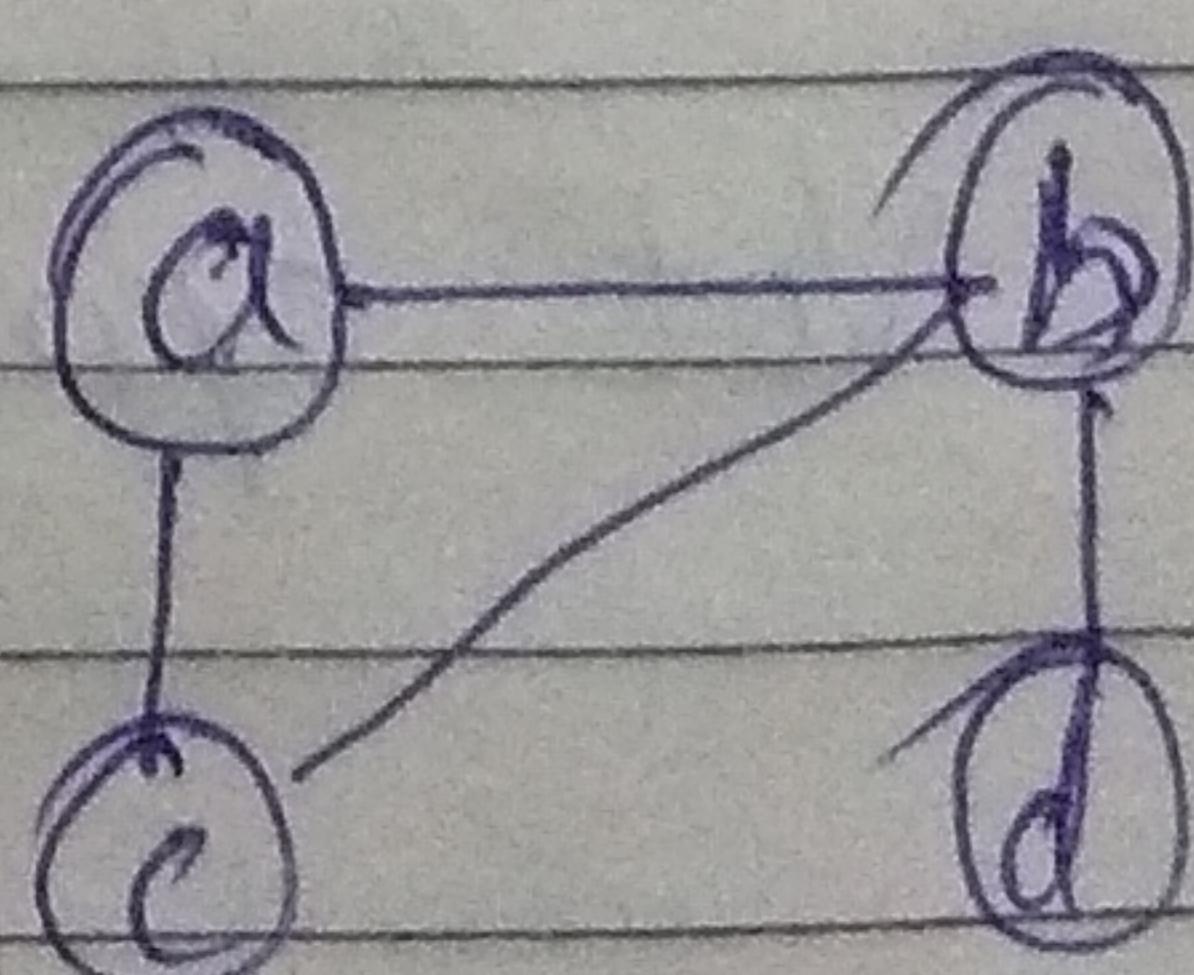


DFS

- ① Create a graph using the given no of edges & vertx.
- ② Create a recursive function that initialize the current index on vertex, visited and recursion stack
- ③ Mark the current nodes as visited & also mark the index in recursion stack.
- ④ find all the vertices which are not visited and are adjacent to the current node
• Recursively call the fun for these vertices
- ⑤ if the adjacent vertices are already marked in the recursion stack then return true
- ⑥ Create a wrapper class, that calls the recursive fun for all the vertices & if any function return true then return true else false.
- ⑦ The diff set can be defined as the subsets where there is no common element between two sets.
 - making new set
 - Union
 - Find

⑥ BFS : A → B → C → D → E → F
DFS : A → B → E → C → D → F → G

- ⑦ Connected components ⑧
Vertices - 10



⑧ Topological sort \rightarrow 0-1-2 \rightarrow 3 \rightarrow 4 \rightarrow 5

Dfs \rightarrow 5 \rightarrow 2 \rightarrow 3 \rightarrow 1 \rightarrow 0

4 cannot be reached.

⑨ Yes, heap data structure can be used to create priority queue

Graph algorithm using priority queue

- Dijkstra's algo - to find shortest path in graph
- Prim's algo
- Kruskal's algo.

⑩ min heap - root element is the smallest
max heap - root element is the largest.